Nondestructive defect detection in castings by using spatial attention bilinear convolutional neural network

Article (Accepted Version)

Nondestructive Defect Detection in Castings by Using Spatial Attention Bilinear Convolutional Neural Network

Zhenhui Tang, Engang Tian, *Member, IEEE*, Yongxiong Wang, Licheng Wang, and Taicheng Yang

***Abstract*—X-ray images of castings are widely used in manufacturing for quality assurance. This article investigates the X-ray-image-based defective detection. The main contributions in this article are twofold: first, a new full-image method is proposed to classify defective castings and nondefective ones; and second, by combining two technologies, spatial attention mechanism and bilinear pooling used in deep convolutional neural networks (CNNs), a new spatial attention bilinear CNN is proposed to enhance the representation power of CNN. To validate the above initiatives, extensive experimental studies have been carried out to show the advantages of the new method over a number of existing ones.**

***Index Terms*—Bilinear convolutional neural network (BCNN), nondestructive defect detection, spatial attention, X-ray testing of castings.**

I. Introduction

AN ADVANCED automatic defect detection system is one of the key elements in smart manufacturing [1]–[7]. In the casting industry, X-ray images are commonly used to detect internal faults. Different X-ray penetration rates of an object with unequal density are the underline physical principle of the X-ray testing. These rates are recorded in an image. As shown in Fig. 1, a typical X-ray testing system consists of the following:

Manuscript received July 23, 2019; revised January 9, 2020 and February 24, 2020; accepted March 29, 2020. Date of publication *******; date of current version *******. This work was supported in part by the National Natural Science Foundation of China under Grant 61773218, Grant 61703245, Grant 61903252, and Grant 61673276; in part by the China Postdoctoral Science Foundation under Grant 2016M600547; in part by the Jiangsu Natural Science Foundation of China under Grant BK2016156; and in part by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning. *(Corresponding author: Engang Tian.)*

Zhenhui Tang, Yongxiong Wang, and Licheng Wang are with the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China (e-mail: 782382950@qq.com; wyxiong@usst.edu.cn; wanglicheng1217@163.com).

Engang Tian is with the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China, and also with College of Automation Electronic Engineering, Qingdao University of Science and Technology, Qingdao 266061, China (e-mail: tianengang@163.com).

Taicheng Yang is with the Department of Engineering and Design, University of Sussex, BN1 9QT Brighton, U.K. (e-mail: t.c.yang@sussex.ac.uk).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2020.2985159

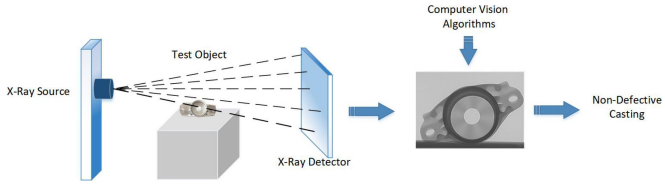

Fig. 1. Sample framework of X-ray testing system of castings.

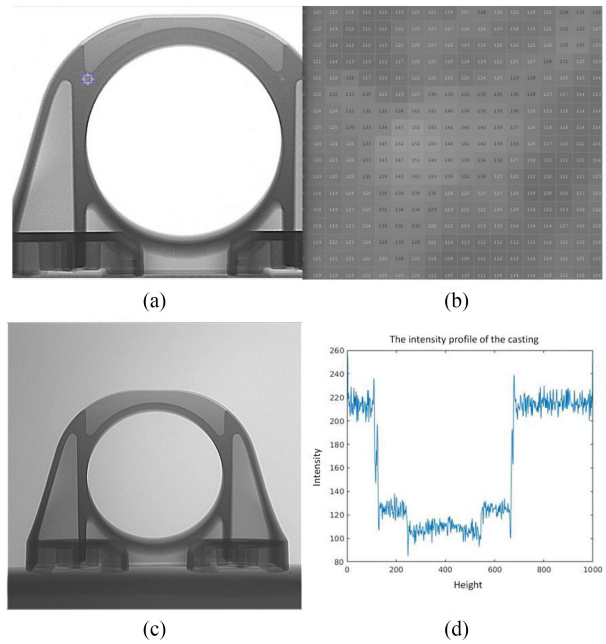

Fig. 2. (a) Sample of the defective castings. (b) Intensity of a defect region. (c) Sample of the nondefective castings. (d) Intensity profile of a casting.

1) the object to be tested, which is fixed in an appropriated position;
2) X-ray source;
3) X-ray detector, which converts X-ray to a digital image;
4) computer vision software to evaluate the X-ray image.

When image processing of X-ray tests is applied, there are two main challenges. First, as shown in Fig. 2, there is noise in the image. The other challenge is that the image of a defect is very subtle and is difficult to distinguish it from the background.

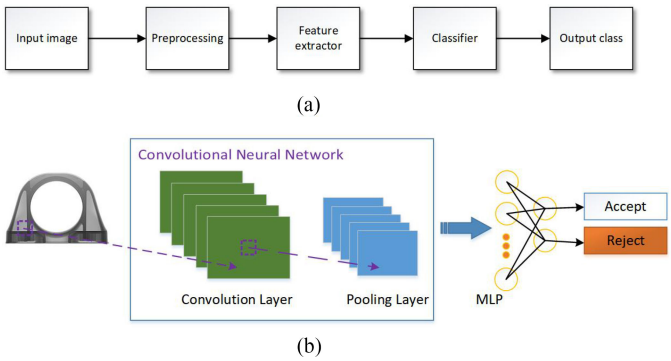

Fig. 3. (a) Traditional computer vision method in the X-ray testing. (b) Pipeline of the CNN for image classification.

Furthermore, defects are often in nonhomogeneous regions such as bubble-shaped voids, fractures, inclusions, or slag formations, which further increase the difficulties of defect detection from an image.

Facing the abovementioned challenges, there are three general approaches [6]–[8], [10] for defect detection: 1) reference image, 2) computer tomography [10], and 3) pattern recognition technologies [6]–[8]. Within the category of pattern recognition, there are two main research directions: object detection and classification. Based on object detection, an object detector is used to classify the defective and nondefective castings. Mery and Arteta [6] evaluated and compared 24 computer vision techniques, including deep learning applied for an automotive component. Ferguson *et al.* [7] proposed the convolutional neural network (CNN) based architectures to localize defects in castings. Based on Mask R-CNN, Ferguson *et al.* [8] proposed a defect detection system to detect and segment defects simultaneously. However, these defect detectors require extra bounding-box labels and a tradeoff between speed and accuracy. Those shortcomings lead to hard implementation. Hence, it is necessary to develop an advanced X-ray testing system.

In our work, the defect detection system is framed as the image classification task. Generally speaking, the image classification of castings or other materials can be classified into two major groups, namely, the classical approaches based on the handcrafted feature extraction [1]–[3], and the deep learning technique based on feature learning, such as CNN [14]–[19]. Mery [9] has reviewed the X-ray testing techniques with computer vision algorithms over the past thirty years. As shown in Fig. 3(a), the traditional image classification method mainly consists of the feature extraction of input image and classification. However, it is difficult to design the right set of features because it needs much prior knowledge and an intensive restriction on the experimental environment.

CNN is one of the most famous deep learning models. It is widely used in image data for automatic feature extraction because of the local connectivity, parameter sharing, translation invariant [13]. Fig. 3(b) shows a classical pipeline of the CNN, which can extract features automatically. Most of the recognition tasks reply on the more effective convolutional features [19]. However, since the visual differences are very small between defective castings and nondefective ones, and process of casting can be affected by many factors such as viewpoint, location, and stuff, how to learn fine-grained features is the core difficulty of this task. Recently, a bilinear model is proposed to mode pairwise feature interaction by computing the gram matrix of feature maps, which has been applied to many image recognition tasks[28], [30] for fine-grained feature learning.

More recently, the visual attention mechanism attracts a lot of research interests. In [24], using clustering from spatially correlated channels, the multiattention CNN has been proposed to localize multiple parts in the feature maps, then features are learned from each part in a mutual reinforcement way. In [26], a spatial and channelwise attention module have been adopted to refine feature maps. Lin *et al.* [28] have proposed a bilinear CNN (BCNN), which can effectively extract texture features and integrate two kinds of features. BCNN is an implicit spatial attention model. It can extract features based on other features. However, since the outer product in [28] aggregates all the spatial position, BCNN ignores the importance of the different spatial region. In essence, the attention mechanism is implemented by assigning large weights to the important regions (i.e., image or feature) and small weights to the unnecessary ones. In this way, representation power is greatly improved.

To overcome the limitation of the BCNN, a new spatial attention module is proposed in this article to explicitly model spatial salience region and improve the classification accuracy. The resulting model is trained end-to-end, and shows better performance through the experiment results.

The main contributions of this article can be summarized as follows.

1) A new CNN architecture (based on spatial attention mechanism and bilinear pooling), namely spatial attention BCNN (SA-BCNN), is proposed for X-ray defect detection for the first time. Our model can classify defective castings directly. And in this way, the label burden can be sharply reduced and more subtle defect difference can be learned, which makes the proposed X-ray defect detection system more practical and suitable for complex tasks.
2) A novel spatial attention module is proposed to reduce the cross-channel correlations and spatial correlations by using depthwise separable convolutions. Experiment results show that our spatial attention module is more efficient.

To illustrate the efficiency of the proposed SA-BCNN, the experiment data is acquired from the real industrial pipelines. In comparison with some prevalent CNNs, the experimental results show that the highest accuracy achieved by SA-BCNN is up to 93.3%. Furthermore, we compare our model with the existing defect detector on Xdefects data set [6]. Experiment results show that our method has obvious performance improvement. For detailed information, please see Tables III and IV. Besides, we also visualize CNN with Grad-CAM to explain the CNN's inner mechanism, which shows that subtle defects can be learned by the CNN.

The rest of this article is organized as follows. Section II presents the proposed method. Section III presents the experimental result. Finally, Section IV concludes this article.

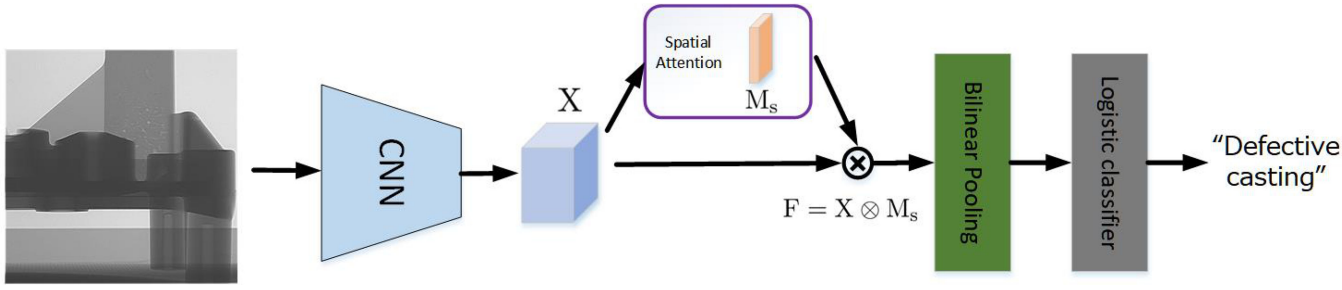

Fig. 4. Architecture of BCNN with spatial attention consists of VGG16, spatial attention, bilinear pooling, and fully connected layer.

TABLE I
NOTATIONS USED IN THIS ARTICLE

Notations	Descriptions
W, m, y	Learnable parameters, number of image, label
$\mathrm{h}, \mathrm{w}, \mathrm{c}$	Height, width, channel
$J(W)$	Loss function
$\mathrm{X} \in \mathrm{R}^{\mathrm{c}\times\mathrm{h}\times\mathrm{w}}$	Feature maps
$\mathrm{x} = \Phi(\mathrm{I})$	Bilinear vector
$\otimes$	Hardamard product
$\mathrm{M_s} \in \mathrm{R}^{1\times\mathrm{h}\times\mathrm{w}}$	Spatial attention map
$\mathrm{F} \in \mathrm{R}^{\mathrm{c}\times\mathrm{h}\times\mathrm{w}}$	Feature maps refined by spatial attention map

II. METHOD

Since the breakthrough result in ImageNet [15], CNN has been widely applied in many computer vision tasks and has achieved outstanding performance. However, it is reported in [24] and [28] that CNN performs poorly in the fine-grained image classification, which obstructs the application of CNN in defect detection. Since the defects of castings are nonhomogeneous regions thus hard to be detected, and visual differences between defective castings and nondefective castings are small, classifying defective castings is still a challenging work.

To deal with this problem, a new CNN architecture based on spatial attention and bilinear pooling (called SA-BCNN) is, for the first time, proposed to learn more discriminative feature. As shown in Fig. 4, the architecture of SA-BCNN consists of the CNN, the spatial attention, the bilinear pooling, and the fully connected layer. It works as follows: first, the origin image is fed into a CNN, which represents an image as feature maps. Second, the feature maps are refined by a spatial attention map, which is calculated based on depthwise separable convolutions. Then, the refined feature maps are converted to bilinear vector through bilinear pooling. At last, the improved bilinear vector is fed into a fully connected layer.

Based on the developed model, in what follows, the traditional CNN architecture is first introduced in Section III-A, and the spatial attention is introduced in Section III-B. Finally, the bilinear pooling is discussed in Section III-C. Some necessary notations appearing in this article are shown in Table I.

A. Convolutional Neural Network

As shown in Fig. 3(b), a basic CNN consists of convolutional layers and pooling layers [14]. Compared with the neural networks, the main characters of CNN include translation invariant, sparse connectivity and parameter sharing. In a CNN, the convolutional operation can extract local pattern in a transitionally invariant manner, and the computation complexity of the convolutional layer [22] is $O(n_{l-1} \cdot k_l^2 \cdot n_l \cdot m_l^2)$, where n_{l-1} and n_l are the output channel number of feature maps in the $(l-1)$th layer and lth layer, respectively, k_l is the kernel size and m_l is the spatial size of the output feature map in the lth layer.

Another important operation of CNN is pooling, which aggregates local information through max operation or mean operation. It makes convolution operation extract features more robust and reduces computational cost observably. Because it can not only reduce the computational cost but also filter the noises, pooling operation is widely used in various CNN architectures. In a classical CNN pipeline, the input image is first represented as feature maps and which is the output of the CNN. And then, the feature maps are fed into fully connected layers and a classifier (i.e., logistic regression, softmax regression). Followed by [23], the classifier in this article is adopted as the logistic regression. The parameters of networks are updated by minimizing the following binary cross entropy (1), where $J(W)$ denotes the loss function, and $W, m, y, \mu(\cdot)$ denote the overall parameters, number of image, label, and Bernoulli distribution, respectively

$$J(W) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)} \log \mu(x^{(i)}; W) + (1 - y^{(i)}) \log(1 - \mu(x^{(i)}; W))]. \quad (1)$$

B. Spatial Attention

Feature maps can be viewed as spatial features. In [26], a spatial attention module is proposed to generate spatial attention map, which uses max-pooling operations and mean-pooling operation to aggregate channel information. However, those pooling operations can lose important channel information and are inefficient. To avoid this problem, we mainly focus on learning the richer spatial representation by reducing channel and spatial correlations.

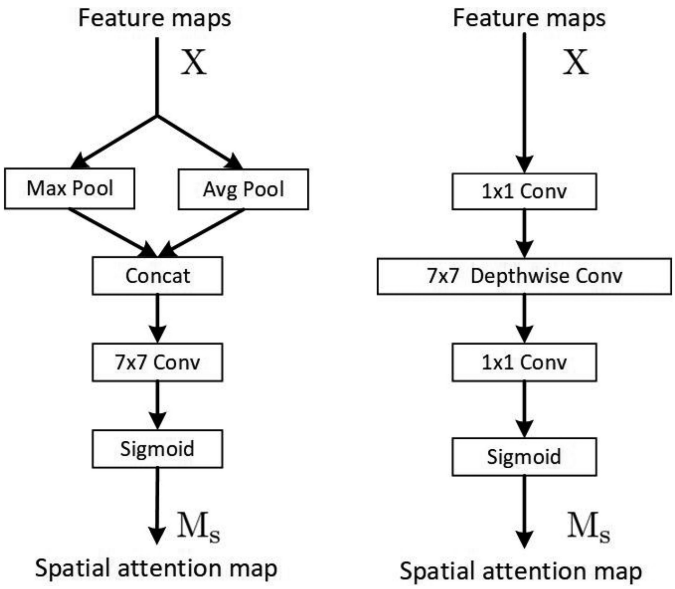

Fig. 5. Left: spatial attention map proposed in CBAM with max-pooling and average-pooling operation followed by a convolutional layer and sigmoid. Right: spatial attention map proposed in this article with 1×1 convolutional layer and depthwise separable convolutional layer followed by 1×1 convolutional layer and sigmoid.

Inspired by Xception [17], a novel spatial attention module is proposed, for the first several attempts, to reduce cross-channel correlations and spatial correlations by using depthwise separable convolutions. In this module, the following conditions hold.

1) A 1×1 convolution is used to learn channel attention correlation and dimensionality reduction.
2) A depthwise separable convolutional layer is used to reduce cross-channel correlations and spatial correlations.
3) A 1×1 convolution is used to generate spatial attention map $\mathrm{M_s}$.

The computational process of spatial attention map can be seen in Fig. 5, which contrasts the spatial attention module in [26] with the proposed spatial attention module.

The refined feature maps are obtained by performing Hardamard product, $\mathrm{F} = \mathrm{X} \otimes \mathrm{M_s}$. The theoretical time complexity of spatial attention module is $O(n_0 \cdot n_1 \cdot m_1^2 + n_1 \cdot k_1^2 \cdot n_2 \cdot m_2^2 + n_2 \cdot n_3 \cdot m_3^2)$.

C. Bilinear Pooling

Bilinear models are effective models to build two key factor variations (i.e., style and content) for images [27]. It has been applied to many tasks including the visual question answer [29], [30], fine-grained recognition [28], and so on. In [28], bilinear pooling is proposed to compute the pairwise feature interactions, which consists of two feature extractors, a pooling function and normalization for the pooled features. To reduce the number of parameters, we use two fully shared CNN as feature extractors. Pooling function is the sum pooling to aggregate all spatial information.

Considering the reshaped feature maps refined by spatial attention map $\mathrm{F} \in \mathrm{R}^{\mathrm{c} \times \mathrm{hw}}$, the bilinear vector $\Phi(\mathrm{F}) \in \mathrm{R}^{\mathrm{c} \times \mathrm{c}}$ is obtained as follows:

$$\Phi(\mathrm{F}) = \frac{1}{hw} \begin{bmatrix} F_1F_1^T & F_1F_2^T & \cdots & F_1F_c^T \\ F_2F_1^T & F_2F_2^T & \cdots & F_2F_c^T \\ \vdots & \vdots & \ddots & \vdots \\ F_cF_1^T & F_cF_2^T & \cdots & F_cF_c^T \end{bmatrix}_{c \times c} = \frac{1}{hw} FF^T \tag{2}$$

where $\mathrm{F_i}\{\mathrm{i} \in 1 \ldots \mathrm{c}\}$ denotes the ith row vector of F, and bilinear vector $\mathrm{x} = \Phi(\mathrm{F})$ is flattened and normalized by a signed square root $(\mathrm{x} \leftarrow \mathrm{sign}(\mathrm{x})\sqrt{|\mathrm{x}|})$ and L_2 normalization $(\mathrm{z} \leftarrow \mathrm{y}/||\mathrm{y}||_2)$. The theoretical time complexity of bilinear pooling is $O(c^2 \cdot hw)$.

As we can see from (2), $\mathrm{F_i F_j^T}$ aggregates all spatial information, thus bilinear pooling can be effectively improved by spatial attention explicitly.

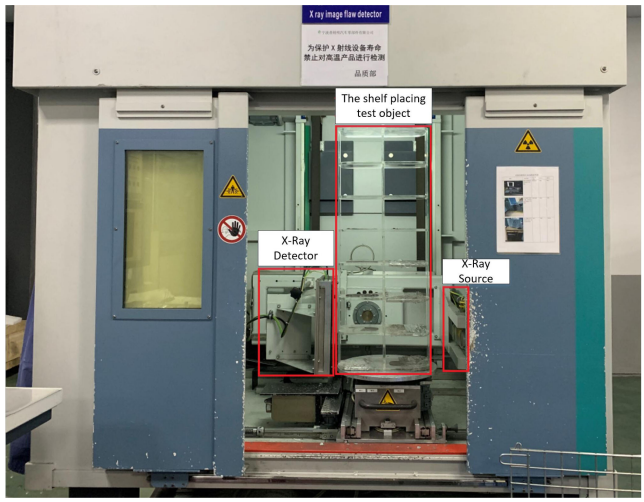

Fig. 6. Casting defect detection system.

III. Experiments

To show the effectiveness of the proposed model in X-ray testing, several experiments are designed and the experimental results are analyzed in detail.

First, an X-ray testing system and experiment data set are discussed in Section III-A. The experiment details and evaluation protocol are introduced in Section III-B and some comparison results with existing CNN are shown in Section III-C. In Section III-D, we also compare the proposed model with some existing defect detectors on a public X-ray data set. Finally, network visualization is conducted by using Grad-CAM in Section III-E.

A. X-Ray Testing System of Castings and Experimental Data Set

As shown in Fig. 6, the X-ray testing system of castings is built to obtain the experimental data set. It consists of the following three modules:

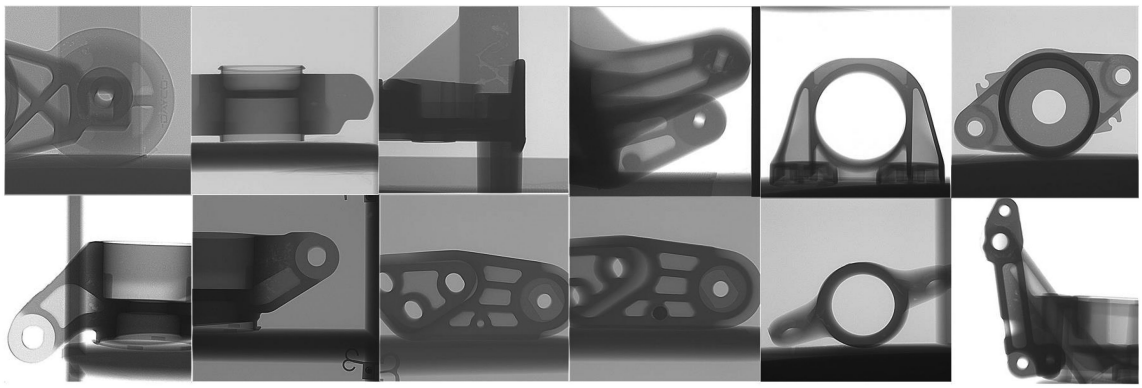

Fig. 7. Several defective castings in the experimental data set.

TABLE II
LIST OF EXPERIMENTAL DATA SET

	Train set	Validation set	Test set
Defective castings	3496	500	500
Non-defective castings	3496	500	500
Total	6992	1000	1000

1) X-ray source, which releases X-ray to castings;
2) shelf, which places the test object;
3) X-ray detector, which collects the energy of X-ray that penetrate the castings while imaging the inner of castings.

The acquired images are gray-value images with the size of 1000×1000. It is an imbalanced data set and includes 59 406 normal castings and 4496 defective castings. To simplify the experiment process, a balanced data set is constructed. As shown in Table II, the experimental data set is split into following:

1) training set with 3496 nondefective images and 3496 defective images;
2) validation set with 500 nondefective images and 500 defective images;
3) test set with 500 nondefective images and 500 defective images.

Several defective castings are shown in Fig. 7.

B. Experiment Details and Evaluation Protocol

For a fair comparison with the state-of-the-art models, we evaluate our model with VGG16 baseline model pretrained on ImageNet and remove the last three fully connected layers. It is worth noting that our method can be combined with other baselines, such as VGG19 and ResNet. For other detailed information, the size of the input image is set as 448×448, and the whole network is trained by using SGD with 50 epoch, batch size of 16, and the learning rate of 10^{-3}. In order to evaluate the performance of the different defective detectors, the following indicators are measured based on the testing set.

1) Params: The total number of the learnable parameters in the model.
2) FLOPs: The total number of the floating point operations of a model.
3) True positive (TP): The number of the defective castings correctly classified.
4) True negative (TN): The number of the defective castings classified as nondefective castings.
5) False positive (FP): The number of the nondefective castings correctly classified.
6) False negative (FN): The number of the nondefective castings classified as defective castings.
7) Precision (Pr): $\frac{\text{TP}}{\text{TP+FP}}$
8) Recall (Re): $\frac{\text{TP}}{\text{TP+FN}}$
9) Accuracy (Acc): $\frac{\text{TP+TN}}{P+N}$
10) Miss (Miss detection rate): $\frac{\text{FN}}{\text{FN+TP}}$
11) False (False alarm rate): $\frac{\text{FP}}{\text{TP+FP}}$
12) CPU: Inference time on the CPU.
13) GPU: Inference time on the GPU.

All experiments are implemented with PyTorch [21] framework and performed on a PC with GTX 1080 and Xeon(R) Silver 4116 CPU.

C. Comparison With Some Prevalent CNNs

To show the efficiency of the proposed method, we compare our model with several prevalent CNNs. Because there are too many layers of the model's fully connected neural network, the model is easy to overfit. To improve it, we replace all models with only one fully connected layer instead. Some of the used models are described in the following.

VGG16 [18]: Very deep CNN, which includes 16 weight layers, 13 convolutional layers, and three fully connected layer. In the comparison, the last three layers are replaced by one fully connected layer.

Xception [17]: A depthwise separable CNN, which includes 36 convolutional layers and one fully connected layer.

ResNet18/34 [19]: A deep residual neural network, which includes 18 convolutional layers or 34 convolutional layers and one fully connected layer.

BCNN [28]: In our experiments, we just consider fully shared BCNN, which uses one CNN (VGG16) as feature extractor.

SA-BCNN (CBAM): BCNN is strengthened by spatial attention module proposed in [26].

TABLE III
COMPARISON EXPERIMENT RESULTS WITH PREVALENT CNNS

Models	Params (M)	FLOPs	CPU(ms)	GPU(ms)	TP	TN	FP	FN	Miss (%)	False (%)	Pr (%)	Re (%)	Acc (%)
VGG16 [18]	14.81	62.52×10^9	444 ±3.98	2.82 ±0.86	458	434	66	42	8.40	12.60	87.40	91.60	89.20
ResNet18 [19]	11.27	7.28×10^9	104 ±1.22	6.92±0.23	429	422	78	71	14.20	15.39	84.61	85.80	85.10
ResNet34 [19]	21.38	14.69×10^9	164 ± 4.23	3.70 ±0.69	423	414	86	77	15.40	16.90	83.10	84.60	83.70
Xception [17]	20.80	18.42×10^9	367 ±2.42	6.09±0.01	462	467	33	38	7.60	6.67	93.33	92.40	92.90
BCNN [28]	14.81	51.38×10^6*	461 ± 26.6	16.80 ±0.01	466	453	47	34	6.80	9.17	90.83	93.20	91.90
SA-BCNN (CBAM)	14.97	71.85×10^6*	481 ± 24.8	17.00 ±0.02	464	461	39	36	7.20	7.76	92.24	92.80	92.50
Our model	15.26	112.79×10^6*	476 ± 18.9	21.40 ±0.02	466	467	33	34	6.80	6.27	93.38	93.20	93.30

*The values with * are only additional plugged modules, the actual FLOPs should add 62.52×10^9.

TABLE IV
EXPERIMENT RESULTS ON XDEFECTS DATA SET

Method	Classifier	TP	TN	FP	FN	Pr(%)	Re(%)	Acc(%)
ImageXnet	SoftMax	5686	7335	210	1850	96.59	75.45	86.39
Xnet	SoftMax	5523	7348	188	2013	96.71	73.29	85.40
VGG-2048	KNN-3	3608	6546	990	3928	78.47	47.88	63.37
VGG-F	KNN-5	3172	6861	675	4364	82.45	42.09	66.57
VGG-19	KNN-3	3263	6644	892	4273	78.53	43.30	65.73
AlexNet	KNN-3	3063	6868	668	4511	81.91	40.14	65.64
GoogleNet	ANN-15	2458	6899	637	5078	79.42	32.62	62.08
BCNN	Logistic	6107	7318	218	1429	96.55	81.04	89.07
SA-BCNN (CBAM)	Logistic	6421	7291	245	1115	96.32	85.20	90.98
Our model	Logistic	6878	6994	542	658	92.70	91.27	92.04

Table III summarizes the experimental results on a test set. It is obvious that our model outperforms all baselines, which demonstrates that the proposed spatial attention module can improve the representation power of bilinear pooling. Compared with spatial attention in [26], the proposed approach is better and efficient as the result of reducing cross-channel correlations and spatial correlation. It is worth mentioning that the accuracy of BCNN, SA-BCNN (CBAM), and our model on train set is 92.7%, 93.4%, and 94.5%, respectively.

It should be pointed out that our model uses VGG16 [18] as a backbone, so the computation complexity (FLOPs) and computation times of the proposed model are larger than those in VGG16 [18] and BCNN [28]. However, it can be seen from Table III that the extra additional computation is very small. Especially, the extra increased computation complexity (FLOPs) can be omitted. Moreover, the performance (see the last five columns in Table III) can be improved obviously. Similarly, when compared with the models in [17] and [19], the proposed model has used a bit more computation time to achieve a better performance.

Furthermore, it is noticed that the CPU time of the proposed model is smaller than SA-BCNN (CBAM), whereas the GPU time is larger than SA-BCNN (CBAM). A similar phenomenon appears in the model Xception [17] and VGG16 [18]. As have explained in [17], the reasons are that the depthwise separable convolution is computationally scattered, and the existing engineering implementation is not optimized on GPU.

Fig. 8 shows the precision–recall curve and miss detection–false alarm curve of various models on testing. From this figure, it can be concluded that compared with the other six models, the proposed model has better precision, lower miss detection rate, and false alarm rate.

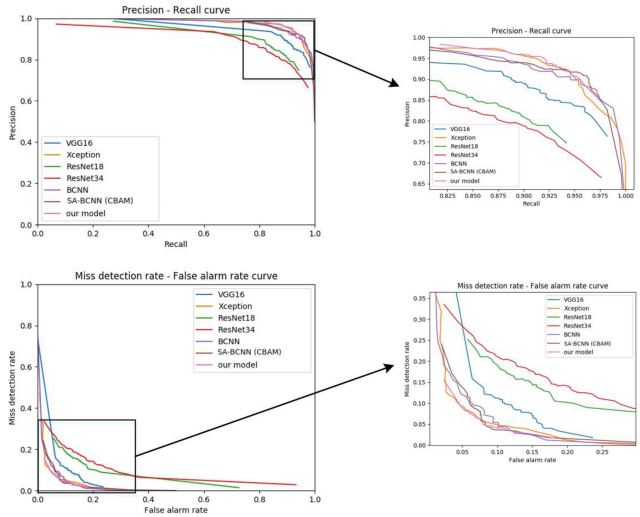

Fig. 8. Precision–recall curve and miss detection–false alarm rate curve.

D. Comparison With Existing Defect Detectors

In this part, some experiments are conducted based on Xdefects data set [6]. The whole data set is consisted of total 47 520 X-ray images including 23 760 defects and 23 760 nondefects, respectively. In our experiment, 32 448 images are used for training and 15 072 images are used for testing. The evaluation protocol is similar to the one in [6]. For preprocessing, images are normalized by min–max normalization and scaled to 448×448.

The experiment results of the proposed model and some popular defect detectors are shown in Table IV. Compared with the deep learning model in [6], it has a significant performance

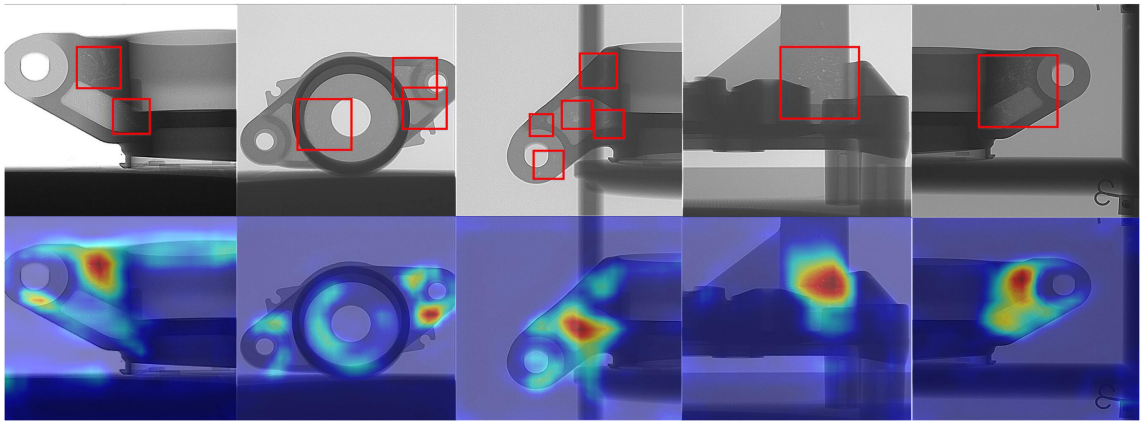

Fig. 9. Network visualization with Grad-CAM.

improvement. For example, SA-BCNN can consistently improve the accuracy up to 90.98%, and our model surpasses BCNN and SA-BCNN (CBAM) by 2.7% and 1.06%, respectively. This shows that our module can learn the discriminative features effectively.

E. Network Visualization With Grad-CAM

For a qualitative analysis, we apply Grad-CAM [31] to VGG16 on the image of the test data set. Grad-CAM is a method to explain the decision from a deep learning network, which can generate a heatmap for a special class on each spatial localization of input image. By using Grad-CAM, which spatial region contributes to the final classification decision can be observed. However, it should be pointed out that Grad-CAM is not suitable for BCNN. In this article, we use Grad-CAM to VGG16 without bilinear pooling and spatial attention model. In Fig. 9, it can be clearly seen that the Grad-CAM masks of VGG16 can cover the defect region coarsely.

IV. Conclusion

In this article, we presented an SA-BCNN to classify defective and nondefective castings. In the proposed model, the spatial attention model and bilinear pooling were integrated, for the first time, into a CNN, which could be trained in an end-to-end manner and extract subtle visual difference. Moreover, a new spatial attention module was proposed by using depthwise separable convolution to reduce cross-channel correlations and spatial correlations. Compared to the proposed SA-BCNN to some prevalent CNNs and existing defect detectors on Xdefects, the experiment results showed that our model have a better detection performance. In addition, we also visualized CNN with Grad-CAM, which showed that subtle defects can be learned by CNN.

Since CNN needs many training examples, its performance will improve with the increase of data. However, the cost of industrial data is much heavy, how to detect defects with the few data will be one of our future research interests.

References

[1] J. Sun, C. Li, X. Wu, V. Palade, and W. Fang, "An effective method of weld defect detection and classification based on machine vision," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6322–6333, Dec. 2019.

[2] J. Wang, Q. Li, J. Gan, H. Yu, and X. Yang, "Surface defect detection via entity sparsity pursuit with intrinsic priors," *IEEE Trans. Ind. Inform.*, vol. 16, no. 1, pp. 141–150, Jan. 2020.

[3] Z. Zhang, G. Wen, and S. Chen, "Audible sound-based intelligent evaluation for aluminum alloy in robotic pulsed GTAW: Mechanism, feature selection, and defect detection," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 2973–2983, Jul. 2018.

[4] S. Lu, J. Feng, H. Zhang, J. Liu, and Z. Wu, "An estimation method of defect size from MFL image using visual transformation convolutional neural network," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 213–224, Jan. 2019.

[5] E. Tian, Z. Wang, L. Zou, and D. Yue, "Chance-constrained H∞ control for a class of time-varying systems with stochastic nonlinearities: The finite-horizon case," *Automatica*, vol. 107, pp. 296–305, 2019.

[6] D. Mery and C. Arteta, "Automatic defect recognition in X-ray testing using computer vision," in *Proc. IEEE Winter Conf. Appl. Comput. Vision*, 2017, pp. 1026–1035.

[7] M. Ferguson, R. Ak, Y. T. Lee, and K. H. Law, "Automatic localization of casting defects with convolutional neural networks," in *Proc. IEEE Int. Conf. Big Data*, 2017, pp. 1726–1735.

[8] M. Ferguson, R. Ak, Y. T. Lee, and K. H. Law, "Detection and segmentation of manufacturing defects with convolutional neural networks and transfer leaning," *Smart Sustain. Manuf. Syst.*, vol. 2, no. 1, pp. 137–164, 2018.

[9] D. Mery, *Computer Vision for X-Ray Testing*. New York, NY, USA: Springer, 2015.

[10] S. Carmignato, W. Dewulf, and R. Leach, *Industrial X-Ray Computed Tomography*. New York, NY, USA: Springer, 2018.

[11] D. Mery *et al.*, "GDXray: The database of X-ray images for nondestructive testing," *J. Nondestruct. Eval.*, vol. 34, no. 4, pp. 34–42, 2015.

[12] C. Miao *et al.*, "SIXray: A large-scale security inspection X-ray benchmark for prohibited item discovery in overlapping images," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 2114–2123.

[13] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[14] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[16] K. Simonyan, and A. Zisserman, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 1–9.

[17] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1800–1807.

[18] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.
[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 15, 1997.
[21] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.
[22] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 5353–5360.
[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Loss functions for binary class probability estimation and classification: Structure and applications," Working Draft, Nov. 2005.
[24] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 5219–5227.
[25] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 7132–7141.
[26] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vision*, 2018, pp. 3–19.
[27] J. B. Tenenbaum and W. T. Freeman, "Separating style and content with bilinear models," *Neural Comput.*, vol. 12, no. 6, pp. 1247–1283, 2000.
[28] T. Lin, A. RoyChowdhury, and S. Maji, "Bilinear convolutional neural networks for fine-grained visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1309–1322, Jun. 2018.
[29] J. H. Kim, J. Jun, and B. T. Zhang, "Bilinear attention networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1564–1574.
[30] Z. Yu, J. Yu, J. Fan, and D. Tao, "Multi-modal factorized bilinear pooling with co-attention learning for visual question answering," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 1839–1848.
[31] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 618–626.
[32] S. Ren, K. He, G. Ross, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[33] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

Zhenhui Tang received the B.Sc degree in electronic information engineering from Soochow University, Suzhou, China, in 2016. He is currently working towards the M.Sc. degrees in control science and engineering at University of Shanghai for Science and Technology, Shanghai, China.

His current research interests include deep learning and its application in computer vision.

Engang Tian (Member, IEEE) received the B.Sc. degree in mathematics from Shandong Normal University, Jinan, China, in 2002, the M.Sc. degree in operations research and cybernetics from Nanjing Normal University, Nanjing, China, in 2005, and the Ph.D. degree in control theory and control engineering from Donghua University, Shanghai, China, in 2008.

From August 2008 to August 2018, he was an Associate Professor and then a Professor with the School of Electrical and Automation Engineering, Nanjing Normal University. He is currently a Professor with the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China. He has authored/coauthored about 80 papers in refereed international journals. His current research interests include networked control systems, as well as nonlinear stochastic control and filtering.

Yongxiong Wang received the B.Sc. degree in engineering mechanics from Harbin Engineering University, Harbin, China, in 1991. And the M. Sc and Ph.D degrees in control science and control engineering from Shanghai Jiao Tong University, Shanghai, China, in 2006 and 2012respectively.

He is currently a Professor in University of Shanghai for Science and Technology, Shanghai, China. His current research interests include computer vision and intelligent robot.

Licheng Wang received the B.Sc. degree in automation from Weifang University, Weifang, China, in 2011, and the M.Sc. and Ph.D. degrees in control science and control engineering from the University of Shanghai for Science and Technology, Shanghai, China, in 2014 and 2019, respectively.

Since November 2016, he has been a Visiting Ph.D. Student with the Department of Electronics and Computer Engineering, Brunel University London, Uxbridge, U.K. He is currently a Postdoctoral Research Fellow with the Department of Control Science and Engineering, University of Shanghai for Science and Technology. His current research interests include nonlinear stochastic control and filtering, as well as complex networks and sensor networks.

Dr. Wang is currently a Reviewer for some international journals.

Taicheng Yang received the Ph.D. degree in electrical engineering from the Control System Centre, University of Manchester Institute of Science and Technology, Manchester, U.K., in 1987.

In 1990, he joined the University of Sussex, Brighton, U.K., as a Lecturer and became a Reader in Computer and Control Engineering in 1999. He had ten years of industrial experience before his academic career. His current research interests include networks and control, power system control and wind power generation, and control theory and applications in general.